TNO report
FEL-96-A078

Time Synchronization in Distributed Simulations

TNO Physics and Electronics Laboratory

TNO

TNO report
FEL-96-A078

Time Synchronization in Distributed Simulations

TNO Physics and Electronics Laboratory

Oude Waalsdorperweg 63
PO Box 96864
2509 JG The Hague
The Netherlands

Phone +31 70 374 00 00
Fax +31 70 328 09 61

Date
May 1996

Author(s)
H.A.M. Luiijf
R. van Kampen

Classification

Classified by	:	F.P. Remmerswaal
Classification date	:	8 May 1996

Title	:	Ongerubriceerd
Managementuittreksel	:	Ongerubriceerd
Abstract	:	Ongerubriceerd
Report text	:	Ongerubriceerd
Appendix A	:	Ongerubriceerd

Copy no	:	9	
No of copies	:	29	
No of pages	:	34	(incl appendix, excl RDP & distribution list)
No of appendices	:	1	

All information which is classified according to Dutch regulations shall be treated by the recipient in the same way as classified information of corresponding value in his own country. No part of this information will be disclosed to any party.

The classification designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret.

The TNO Physics and Electronics Laboratory is part of TNO Defence Research which further consists of:

TNO Prins Maurits Laboratory
TNO Human Factors Research Institute

Netherlands Organization for Applied Scientific Research

Managementuittreksel

Titel	:	Time Synchronization in Distributed Simulations
Auteur(s)	:	Ir. H.A.M. Luiijf, Ir. R. van Kampen
Datum	:	mei 1996
Opdrachtnr.	:	A95KL840
IWP-nr.	:	756
Rapportnr.	:	FEL-96-A078

Bij het koppelen van gedistribueerde simulaties dienen de interne klokken van deze simulatoren gesynchroniseerd te worden om de gesimuleerde realiteit zo goed mogelijk te presenteren, door middel van timestamping de effecten van netwerk- en andere vertragingen te compenseren èn de causaliteit van opeenvolgende gebeurtenissen te waarborgen. Met name is dit van belang bij de koppeling van simulatoren via een wide-area netwerk met gemiddelde vertragingen in de gegevensuitwisseling in de orde van enkele honderden milliseconden.

Middels experimenten tussen het Institute for Simulation and Training (IST) en Veda Inc., beide gevestigd in Orlando, Florida en TNO-FEL is onderzocht:
- op welke wijze tijdsynchronisatie tussen gedistribueerde simulaties kan plaatsvinden,
- welke wijze van tijdsynchronisatie de voorkeur verdient,
- welke klokstabiliteit verkregen kan worden,
- in hoeverre de netwerkbelasting hierop van invloed is,
- wat de gemiddelde vertragingstijd en -variantie is,
- op welke plaatsen in de protocol-stacks welke vertraging optreedt,
- wat de invloed van al deze variabelen is op natuurgetrouwheid van de simulaties.

Daarnaast is de wijze waarop NTP geïnstalleerd en geconfigureerd moet worden in dit rapport beschreven. Dit omdat de met NTP meegeleverde documentatie het installatie- en tuning-proces onvolledig beschrijft.

Voor de koppeling van de DIS-systemen op de drie locaties is met veel succes gebruik gemaakt van het Integrated Services Digital Network (ISDN).
Tijdens de experimenten is Internet Relay Chat (IRC) een zeer (kosten)effectief hulpmiddel gebleken om de communicatie tussen gedistribueerde samenwerkende partners te onderhouden.

Conclusies die de experimenten opgeleverd hebben:
- Voor DIS-toepassingen is ISDN een uitstekende manier om wide-area koppelingen tot stand te brengen gezien de eenvoudige wijze van koppelen en de lage variantie in de netwerk-latency.

- Klokken van standaard computersystemen verlopen wel 2 seconden per 6 tot 7 minuten. Simulaties die van deze klokken gebruik maken, kunnen dit verlopen van klokken niet compenseren.
- Het Network Time Protocol (NTP) levert in combinatie met een stabiele klokbron als bijvoorbeeld een GPS-klok na enige 'systeemtuning' een zeer goede tijdsynchronisatie tussen simulatoren op, ook via een zwaar belast wide-area netwerk.
- Om te compenseren voor het verlopen van systeemklokken is het nodig om het verloop van een systeemklok te kunnen meten. Hiervoor is NTP een uitstekend hulpmiddel.
  In dit licht gezien wordt het gebruik van timestamping op basis van relatieve tijd in DIS afgeraden en wordt het gebruik van absolute timestamping met op UTC (bijv. GPS) gesynchroniseerde klokken aangeraden.
- Voor samenwerkingsprojecten tussen twee of meer (inter)nationale partners, waarbij momentane uitwisseling van informatie van belang is voor experimenten of metingen, wordt het gebruik van Internet Relay Chat aangeraden.

De experimenten vonden plaats in het kader van de Data Exchange Agreement tussen de US AMC/STRICOM en de DMKL.

Contents

1. Introduction

As part of the Mutual Weapons Development Master Data Exchange Agreement between the US AMC/STRICOM and the RNlA/DMKL (DEA Annex A-94-TN-1529), experiments were conducted between the Institute for Simulation and Training (IST), TNO-FEL, and Veda Inc. in the implementation and analysis of time synchronization among the three DIS-sites connected by an Integrated Services Digital Network (ISDN) Wide Area Network.

Each site implemented a Global Positioning System (GPS) receiver as a local source of UTC, and used the GPS receiver to synchronize via two approaches: the Network Time Protocol (NTP) and via SIMAN PDUs.

When simulation applications exchange information across a network, it is often necessary to compensate for the latencies involved. This is especially true in a Wide Area Network (WAN) with average transport latencies in the hundreds of milliseconds. A distributed simulation will usually have a requirement for fidelity of ground truth perception that may or may not be achievable through measures to compensate for WAN latencies. The degree of compensation that may be achieved is dependent on several key variables, including synchronization to Coordinated Universal Time (UTC), stabilization of system clocks, network loading, average latency, latency variance, and the fidelity of the simulation applications.

The research focused on two areas: examining the characteristics of Integrated Services Digital Networks (ISDN), and conducting DIS experiments using absolute timestamps across a Wide Area Network (WAN). Each of these areas are relevant to the DIS community. ISDN is potentially a valuable asset as a primary or backup communications link, offering relatively high bandwith, increasing availability and low costs, whereas the use of absolute time is a fundamental yet somewhat unresolved issue in Distributed Interactive Simulation.

Our experiences in setting up an absolute time source and in configuring the Network Time Protocol (NTP) are well documented in later chapters, which the authors hope will be seen as a valuable reference for future installation and tuning of NTP-synchronized systems.

Additionally, references from previous DIS workshops are cited which address the key issues of latency, variance, time synchronization, and dead reckoning, and are highly recommended reading to gain a clearer insight into the issues involved in the use of absolute timestamps in DIS.

This contents of this report was presented during the 14th Workshop on Standards for the Interoperability of Distributed Simulations [13] as a joint paper by the organizations mentioned above. The co-authors of the paper and co-workers during the experiments Andy Cox of IST, Orlando, Florida and Rob Ripley, Veda, Inc,

Orlando, Florida should be regarded as co-authors of this report as well. The paper itself has been published in the proceedings of the 14th Workshop on Standards for the Interoperability of Distributed Simulations and is available on Internet as a downloadable file [13].

The support of our sponsors Ir. O. Hoogesteijn (RNLA/DMKL/T&WO) and Mr. Michael P. Garnsey (U.S. Army STRICOM) for this project is gratefully acknowledged.

2. Background

ISDN service was established at each site to provide an on-demand WAN connection. A testbed of simulation applications, including ModSAF, the IST Computer Generated Forces (IST CGF), and Veda's DISMan Simulation Manager were used to provide DIS traffic for analysis. TNO-FEL provided expertise in the use of ISDN as an alternative to T-1 lines, including the use of ISDN during the 1995 ITEC DIS demonstrations [7, 14]. IST and Veda had participated in the development of the Simulation Management (SIMAN) protocol in DIS, including SIMAN demonstrations conducted during the Interservice/Industry Training Systems and Education Conferences (I/ITSEC) in 1994 and 1995, and proposed to investigate the use of SIMAN as a possible approach to time synchronization.

2.1 Coordinated Universal Time (UTC)

Agreeing upon the definition of an absolute time base is an essential element in establishing synchronization. DIS defines absolute time as synchronization to Coordinated Universal Time, or UTC. UTC is a time defined by a collection of atomic clocks, and the abbreviation UTC may be followed by an organization who published the particular time reference signal. For example, the U.S. Naval Observatory (USNO) maintains the USNO Master Clock which consists of dozens of cesium atomic clocks and several hydrogen maser clocks. This USNO Master Clock is the source of UTC(USNO). A definition of absolute time comprises several related terms, provided in the following paragraphs.

Atomic time: the unit of duration the Systeme International (SI) second defined as the duration of 9,192,631,770 cycles of radiation corresponding to the transition between two hyperfine levels of the ground state of cesium 133.

TAI is the International Atomic Time scale, a statistical timescale based on a large number of atomic clocks.

***Universal time** (UT)* is counted from 0 hours at midnight, with unit of duration the mean solar day, defined to be as uniform as possible despite variations in the rotation of the Earth.

UT0 is the rotational timescale of a particular place of observation. It is observed as the diurnal motion of stars or extraterrestrial radio sources.
UT1 is computed by correcting UT0 for the effect of polar motion on the longitude of the observing site. It varies from uniformity because of the irregularities in the Earth's rotation.

Coordinated Universal Time (*UTC*) differs from TAI by an integral number of seconds. UTC is kept within 0.9 seconds of UT1 by the introduction of one-second steps to UTC, the "leap second" usually being a positive step.

Note that UTC differs from TAI by the cumulative number of leap second events that have taken place, currently at 11, the most recent having occurred 31 December 1995 during the course of our experiments. Each site experienced a temporary loss of synchronization as our clocks were set back one second by NTP, representing the change in UTC. NTP reestablished synchronization shortly thereafter. The use of GPS and NTP in our experiments are discussed in later chapters.

2.2 The DIS Timestamp

In the DIS standard (IEEE 1278.1), entities exchange information by issuing Protocol Data Units (PDUs) across a network. Each PDU contains a timestamp, which indicates the time at which the information within the PDU is valid. The timestamp may be either absolute or relative, with absolute timestamps being used when the sending application is synchronized to UTC. It should be noted that the DIS timestamp is represented by a 32-bit unsigned integer representing time past the hour, with the least significant bit used as an absolute/relative flag. The remaining 31 bits are used to represent one hour, so the DIS timestamp rolls over at the end of each hour. The DIS representation seems awkward, oddly representing time in units of 1.66667 microseconds. There is no representation of anything but “time past the hour” in the DIS timestamp. A recommendation was once proposed to change the DIS timestamp to a more “conventional” representation of hours (or seconds) + microseconds past the hour, which would require an additional 32 bits but provide greater accuracy and usability, but the suggestion was never adopted.

The use of absolute timestamps allow other entities to determine precisely how much time has elapsed since a PDU was issued, for each PDU, with accuracy limited only by the synchronization to UTC of the sending and receiving applications. The same precision is not possible in a relative time scheme, which requires estimates based on observed network latencies. An even less precise approach would be to use time of receipt, disregarding timestamps entirely, which avoids the significant errors that can occur due to uncompensated clock drift. Time of receipt is not addressed in the DIS Standard, although it is common practice. [1]

2.3 Site Configurations and Infrastructure

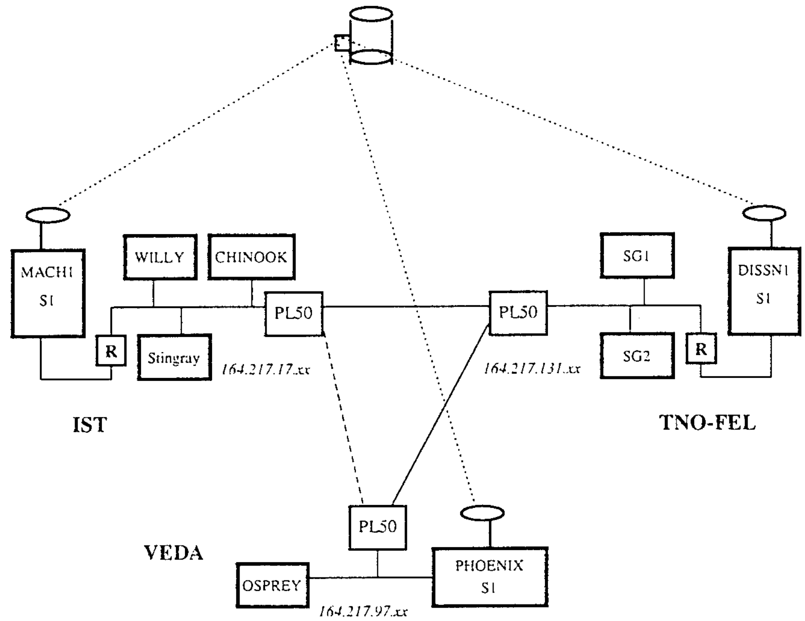

Figure 1: Site Topology

The experiments between IST, TNO-FEL, and Veda were conducted using the site configurations and infrastructure shown in Figure 1:

- Each site has a Global Positioning System (GPS) receiver which receives data from several GPS satellites. Each GPS-receiver provides highly accurate time to an NTP stratum-1 server. These servers are marked S1. See chapter 4 for an explanation of NTP strata.
- Routers ("R") were required at IST and TNO-FEL to access the stratum-1 servers, which resided on a different subnet. The simulation applications and loggers were on an isolated subnet using the 164.217 address and site IDs used during I/ITSEC DIS demonstrations.
- The sites were interconnected using Ascend PL50 ISDN-bridges/routers. There were three possible configurations: two Atlantic ocean crossings (two 'long legs' as shown by the solid lines connecting TNO-FEL and Veda) with the hub point at TNO-FEL, a hub point at either Veda or IST (a short and a long leg), and a triangle (shortest transmission path for each of the sites).
- Machine types were as follows: SG Indy: Chinook, Willy, and Osprey. SG Indigo: SG1, SG2, and Phoenix. Sun Sparc10: Mach-1 and DISSN1. PC: Stingray.
- The simulation applications used for the experiments were ModSAF (SG1), the IST CGF (Willy), and DISMan (Osprey). Chinook, Phoenix, and SG2 were used for data logging.

3. Integrated Services Digital Network (ISDN)

For the planned timing experiments, it was decided to use ISDN in lieu of the DSI to provide a wide area link between IST, TNO-FEL, and Veda. This decision was made in part to provide more insight into the suitability of ISDN for distributed simulation, and to provide "on-demand" connectivity between the sites without the requirements involved in coordinating DSI usage. As reported in [7] and [14], a ISDN-link is in principle just dial and run after the configurations are set up. As stated by TNO-FEL in those reports, ISDN technology is still new and might cause unexpected problems. During our experiments, we experienced the same positive and negative experiences.

The initial, easily fixed problems were:

- entering the correct international ISDN (phone)numbers;
- no international ISDN-calling service contract at one of the sites;
- setting the correct dial-filters and line hang-up time-outs.

Most of this (re)configuration was done remotely from The Netherlands by opening a telnet-session to the PL50. Internet Relay Chat (IRC, discussed in paragraph 5.2) turned out to be invaluable as a communication tool, allowing us to "talk" throughout the experiments.

The experiments were hampered for a couple of days by a major problem, later identified as a bug in the Ascend PL50 software. There were, in fact, three software upgrades required during the course of the experiments to correct various problems.

We then found that UDP broadcast packets weren't forwarded between the two ISDN B-channels on the PL50 acting as the hub. As a temporary work-around, TNO-FEL moved the hub-point to the TNO Ethernet by using a second PL50 for the second B-channel access. A copy of the IST PC CGF was provided to Ascend technical support, which duplicated our PL50 configurations to verify that there was indeed a problem in their system. Shortly thereafter, a work-around was received from Ascend. By restarting the router with no assigned IP address the PL50 would then forward UDP broadcast traffic. However, without an IP address we could not telnet to the router and were required to use a terminal program through the PL50 serial port. The bug will be fixed by Ascend Inc. in a future software release.

Due to these problems, there wasn't time left to experiment much with the triangle filtering and triangle router setups. A triangle bridged network setup was stable for only a brief period, and is sensitive to a good filtering setup. We have seen an instability due to multicast storms of undetermined origin when in the triangle configuration. Thus more work has to be done in preparing a guideline for this type of interconnection.

The ISDN Wide Area Network used in these experiments turned out to be highly reliable and exhibited minimal variance, characteristics which are important in DIS.

4. Network Time Protocol (NTP)

NTP is a mechanism for synchronizing system clocks over local and wide area TCP/IP networks. The current version of NTP is specified in the DARPA Network Working Group document RFC-1305, titled *Network Time Protocol (Version 3) Specification, Implementation and Analysis* [2], written by David Mills of the University of Delaware. In order to properly install, configure and use NTP, one must first have a general understanding of how NTP works. There are many good references available that explain how NTP works (see references in chapter 9) and some go into great detail on the topic. This chapter will attempt to provide enough insight into the general workings of NTP such that the reader will understand how it was implemented and configured for this particular project.
Readers not interested in the inside, installation and tuning of NTP should proceed to paragraph 4.6.

4.1 How NTP Works

Each server in an NTP network "attempts to synchronize to UTC using the best available source and available transmission paths to that source."[4] NTP does not attempt to synchronize computers to each other, but rather it tries to synchronize computer clocks to UTC. NTP assumes that there is one true standard time (UTC) and it maintains relationships with other NTP servers to best determine what UTC is.

Time is distributed through a hierarchy of NTP servers and each server operates at a specified "stratum" level. The stratum level of a given server indicates how far away from an external source of UTC the server operates at. For instance, a server that advertises itself as a stratum-1 server is by definition directly connected to an external source of UTC (such as a UTC(GPS) from a GPS receiver). A stratum-2 server, not having a direct source of UTC, must get UTC from a stratum-1 server. A stratum-3 server gets UTC from a stratum-2 server, and so on. The number of NTP stratum levels is limited to 15. (To avoid confusion, this report will refer to stratum 2 through 15 as being "lower" than stratum-1.)

In order to effectively operate over the non-deterministic path lengths of packet-switched networks, NTP estimates three key variables in the relationship between a client (such as a stratum-2 server) and a timeserver (such as a stratum-1 server): network delay, dispersion of time packet exchanges (a measure of maximum clock error between two hosts), and clock offset, which is the amount of correction that should be applied to a clock in order to synchronize it with UTC. [5] NTP applies estimates of these three variables in order to synchronize a system clock.

One of the major obstacles to synchronizing a UNIX system clock is the amount of error in the clock itself. A UNIX clock consists of an oscillator and a hardware

counter. Using the oscillator, the counter triggers interrupts to the system a certain number of times per second. For most UNIX computers, this happens (by default) 100 times per second, or once every 10 ms. Each time a clock interrupt occurs, the value of the system clock is incremented by a certain value.
On Silicon Graphics Irix computers, this value is determined by the kernel variable *timetrim*; on Sun Solaris computers, this is determined by the kernel variable *nsec_per_tick*. The error creeps into this formula by way of the clock oscillator. An unstable clock oscillator can cause a system clock to run fast, slow, or erratic. In addition, other factors such as system messages written to the console and excessive CPU or I/O usage can cause clock interrupts to be delayed, causing an unstable clock. Fortunately, NTP is smart enough to offset some of these errors. However, a system with a stable clock oscillator will almost always give better clock synchronization results.

4.2 The Xntp Software

Xntp is a UNIX software package that is a complete implementation of the NTP Version 3 specification. For this project, IST and Veda used Xntp version 3.4x and TNO used the newer 3.4y version. The Xntp software distribution is readily available on the Internet and is free of charge. The software can be downloaded via anonymous FTP from *louie.udel.edu* or from the World Wide Web page found at *http://www.eecis.udel.edu/~ntp/*, which also contains links to a number of helpful documents, some of which are listed in chapter 9.

4.3 The Global Positioning System (GPS)

In order to synchronize host clocks on a network, a method of getting accurate time must first be found. The NAVSTAR GPS satellites transmit signals that enable mobile or stationary GPS receivers to determine accurate position and time. GPS receivers are available commercially from a variety of vendors. Receivers range from small, portable models costing in the hundreds of dollars to rack-mounted, highly accurate models costing nearly $10,000. For this project, it was important to find models that were accurate to within a few microseconds, but that were also affordable. Another factor which influenced the decision on which receivers to buy was the level of support offered by Xntp for a particular receiver. The file README.refclock in the doc directory of the Xntp distribution lists the supported receivers.

The Xntp software can read the time code output of a variety of radio and satellite receivers, including GPS receivers. In order to determine which GPS receivers to buy, it was necessary to research the GPS receivers supported by Xntp. After weighing cost, level of NTP support, and accuracy of timing output, IST selected the Trimble Navigation SVeeSix, Veda selected the TrueTime Model GPS-TMS,

and TNO selected the TrueTime Model GPS-TMD. Each of these receivers is fully supported by Xntp and each has a timing accuracy of +/- 2 microseconds. At all sites, the GPS receiver was connected to the host computer via the RS-232 serial port.

GPS clocks output UTC time in an ASCII string that is sent to the NTP server through a serial interface. This should give you approximately millisecond accuracy. GPS clocks also output a PPS (pulse per second) signal. Using the PPS signal should result in 4 microsecond accuracy.

By just using the serial interface, all three of our sites were able to achieve approximately millisecond accuracy. TNO tried to improve upon that accuracy using the PPS clock, but was not able to get the Xntp software to listen to the PPS signal during the course of the experiments. The ppsclock software that comes with the Xntp software distribution works only on Sun workstations running old SunOS 4.1.X.

After the experiments which were presented in the paper to the 14th Workshop on Standards for the Interoperability of Distributed Simulations [13], an old, almost obsolete SUN system was reconfigured at TNO-FEL as a NTP Stratum-1 server using the ppsclock-code as sychronization source to the GPS PPS-signal. A stability within the range of 6 microseconds of UTC has been achieved, a large improvement in clock stability over the already quite stable system clock.

4.4 Configuring the NTP Software for Stratum-1 Servers

Most of the steps to take in configuring a stratum-1 server are generic across different vendors' computers. However, some steps differ among different host platforms and operating systems. The steps to take in configuring the NTP software on a stratum-1 server are outlined below for Silicon Graphics Irix and SUN Solaris computers.

Step 1. Make the NTP daemon (called xntpd) by following the instructions in the file RELNOTES, which can be found in the Xntp distribution.

Step 2. Create the file /etc/ntp.conf with the following contents (modify as necessary for your site).

server 127.127.15.1 minpoll 4 maxpoll 4
driftfile /etc/ntp.drift
statsdir /usr/local/xntp3_4x/scripts/stats/
statistics loopstats
filegen loopstats file loopstats type day

The first line tells xntpd that it is to get UTC from a TrueTime GPS-TMS receiver that is directly connected to the host computer. The addressing scheme (e.g., 127.127.15.1 for a GPS-TMS) is explained in detail in the README.refclock file that is contained in the Xntp distribution. The minpoll and maxpoll options indicate that xntpd is to poll for UTC every 2^4 (or 16) seconds. This is a higher polling rate than xntpd's default for a stratum-1 (64 seconds) and effectively causes xntpd to synchronize faster should it ever get out of sync for any reason.

The second line tells xntpd where to save the "drift" value that it calculates. The drift value is the computed error in the intrinsic frequency of the clock on which xntpd is running. It takes xntpd a day or so to compute this value, but since the value is continually recomputed and saved in the "driftfile", xntpd should never need to recompute it from scratch.

The remaining lines in ntp.conf tell xntpd to generate a new loopstats file once per day in the /usr/local/xntp3_4x/scripts/stats directory (replace this directory name with whatever you wish). The loopstats file displays information about how well xntpd thinks it is synched to UTC. Generating this file is optional, but it is very useful for keeping track of how well xntpd is operating.

Step 3. Assuming the GPS receiver is connected to serial port #1, create a link in the /dev directory called gpstm1 to the serial port #1 device. According to the README.refclock file, xntpd will talk to the GPS receiver via the serial port by looking for a device with the following naming scheme:

/dev/<device name><device number>

In this example for a TrueTime GPS-TMS, the device name is "gpstm" (see README.refclock) and the device number is "1", since it is the first (and only) GPS device. So, enter the following command to create the link:
For SGI Irix:

ln /dev/ttyd1 /dev/gpstm1

For Sun Solaris:

ln /dev/term/a /dev/gpstm1

Step 4. On SGI Irix workstations, edit the file /etc/inittab. Since (in our example) the GPS receiver is connected to serial port #1, find the line that begins with "t1:". Make sure it looks like this:

t1:23:off:/sbin/getty -N ttyd2 dx_9600

The key word here is "off" so that getty won't try to access the serial port.

On Sun Solaris workstations, you will need to enter the following command (assuming the GPS receiver is connected to serial port #1):

pmadm -r -p zsmon -s ttya

Step 5. On SGI Irix workstations, follow the recommendation in the sgi file (distributed with Xntp and located in the hints directory) to set the *duart_rsrv_duration* flag to zero in /var/sysgen/master.d/sduart. This is done to improve input latency on the serial port and is a very important step. Testing has revealed that this step markedly improves the stability of clock synchronization on an SGI.

On Sun Solaris workstations, disable synchronization to the battery clock by setting the kernel variable *dosynctodr* to zero in the file /etc/system.

Step 6. On SGI Irix workstations, disable the timed and timeslave daemons by ensuring that the contents of the files /etc/config/timed and /etc/config/timeslave contain only the keyword "off".

Step 7. The NTP daemon constantly adjusts the system clock. NTP also determines the drift (i.e. the deviation) of the speed of the system clock. The drift is expressed in parts per million (PPM), where a million is 2**20 and not 10**6. (Perhaps we should call it parts per megapart). NTP, however, does not alter the speed of the system clock to compensate for the drift. It is essential that you do that.

On SGI Irix workstations, use the timetrim.c program found in the util directory of the Xntp distribution to set the system clock frequency trim kernel variable called *timetrim* to zero. (Instructions on using the program can be found in timetrim.c.) Next, run xntpd. After xntpd has run for a few hours, check the /var/adm/SYSLOG file to see if xntpd has synchronized. If it has not synchronized and is continuously stepping the time because the clock drift rate is too high, manually calculate an approximate timetrim value to correct the observed drift rate and apply it with the timetrim program. Xntpd should then synchronize. [6] Let xntpd run for a day or two and then check the drift value in /etc/ntp.drift. Using the timetrim program, set *timetrim* to the current drift value. This will effectively cause xntpd to lower the drift value and will help xntpd to stabilize. You may need to repeat this step a few times until the drift value stabilizes around zero. Add the command to set *timetrim* in a system startup file so that *timetrim* has the correct value when xntpd starts up.

On Sun Solaris workstations, the process is similar to that for SGI workstations, but instead of using *timetrim*, you can alter the speed of the system clock to compensate for the drift by setting the *nsec_per_tick* kernel variable in the file /etc/system. Don't forget to remove the ntp.drift file after altering the speed of the system clock. You will probably need to tune the speed of the system clock a couple of times until the drift fluctuates around zero.

Step 8. Add the xntpd start command to a system startup file so that xntpd will be started up at boot time. There are a number of command line options for xntpd (found in the xntpd.8 man page), but it is sufficient to just run it with no command line options.

4.5 Configuring the NTP Software for Stratum-2 (and Lower) Servers

The steps taken to configure a stratum-2 (or lower stratum) server are outlined below. In order to configure a stratum-2, it is necessary to determine which stratum-1 server(s) it will receive time from and which stratum-2 computers it will peer with. The more servers and peers that are identified for a stratum-2, the better are the chances of good time synchronization. The following steps for configuring a stratum-2 server assume the NTP daemon has already been compiled.

Step 1. Create the file /etc/ntp.conf with the following contents:

server <stratum-1 IP addr> minpoll 5 maxpoll 7
peer <stratum-2 peer IP addr>
driftfile /etc/ntp.drift
statsdir /usr/local/xntp3_4x/scripts/stats/
statistics loopstats
filegen loopstats file loopstats type day

In the first line, the "server" keyword indicates that this stratum-2 machine is to receive synchronization from the indicated server (insert the correct IP address of the server as indicated), but will not provide synchronization to the server. The minpoll and maxpoll options (explained earlier) are optional, but were found to be effective at keeping the lower stratum machines in close synchronization. By default, xntpd maintains a constant polling period of 64 seconds for the stratum-1 machines. For the machines at the lower stratum levels, it starts with 64 seconds and doubles its value whenever the algorithm "feels safe" to do so, until it reaches 1024 seconds. Limiting the polling range (in our example, between 2^5 and 2^7 seconds) effectively forces the lower stratum machines to poll at the rate you desire.

The second line, which is optional but recommended, tells xntpd to peer with another machine running NTP at the same stratum level. Machines that maintain a peer relationship will both provide and receive synchronization. This is a good practice that increases the likelihood of stable synchronization.

The remaining lines in /etc/ntp.conf are the same as for the stratum-1 server described earlier.

Step 2. Continue with steps 6 through 8 of the stratum-1 server setup.

4.6 NTP Performance: what we learned

Considering the distances between the sites, our clocks synchronized quite well. The fact that ISDN latencies were fairly stable made it easier on NTP to maintain sync. There were some problems, though. The IST stratum-1 server was consistently about 40 milliseconds behind UTC. The cause is unknown, but is being investigated. As a temporary solution, the IST stratum-2 workstations opted to sync with the Veda stratum-1 and quickly regained sync. Following the NTP hints, TNO entered all the other workstations in our experiment as NTP servers in the NTP configuration file /etc/ntp.conf. This caused the TNO stratum-2 servers to opt for the best NTP server to sync to. The best NTP server, however, would fluctuate during the course of the experiments, causing variations in the offset from TNO's stratum-2 workstations to Veda's stratum-1 server. IST stratum-2 workstations remained within about 2-3 milliseconds to the Veda stratum-1 and TNO stratum-2 workstations stayed within about 6 milliseconds of the Veda stratum-1. These stratum-1 problems point out that for clock synchronization, there is an absolute need for a stable stratum-1 server. Some vendors are introducing all-in-one GPS/NTP stratum-1 boxes that connect directly to a LAN in a plug-and-play fashion. Though fairly expensive, they obviate the need for a dedicated UNIX machine to act as a stratum-1 and they may make setting up a stratum-1 server as easy as plugging it in.

We observed that excessive CPU usage and screen I/O usage could have a negative impact on clock synchronization. Heavy traffic on the network, however, did not seem to effect the NTP servers, nor did disk I/O. The Veda machine Phoenix was both the S1 server and a data logger. The act of logging network traffic to a file did not have a significant impact on the ability to maintain synchronization to the the GPS signal.

We concluded that as long as WAN connections can remain active, only one stratum-1 server with a GPS receiver is necessary to synchronize the clocks at various remote sites in a DIS-excercize. If the WAN-connections are not up all the time, a stratum-1 at each site would be required.

The Xntp software distribution contains two programs that can be used to estimate the offset between system clocks: ntpdate and xntpdc. Ntpdate polls the NTP servers that are specified on the command line. Ntpdate obtains a number of samples from each server to estimate the clock offset in a way similar to the NTP daemon. Ntpdate may be used to set the system clock when run with root privileges. When run in debug mode it queries statistics but does not attempt to set the host clock. It also will fail to set the clock if the ntp daemon xntpd is running. Xntpdc is a program that queries the NTP daemon itself and reports the values that the daemon last calculated. Since ntpdate calculates clock offset "on the fly", we decided ntpdate was the better tool with which to measure how closely our clocks were synchronized. Figures 2 through 6 show clock offsets (measured with ntpdate) from various stratum-2 computers to the stratum-1 server at Veda (which

is node Phoenix). See figure 1 (site topology) to identify the computer names shown in figures 2 through 6.

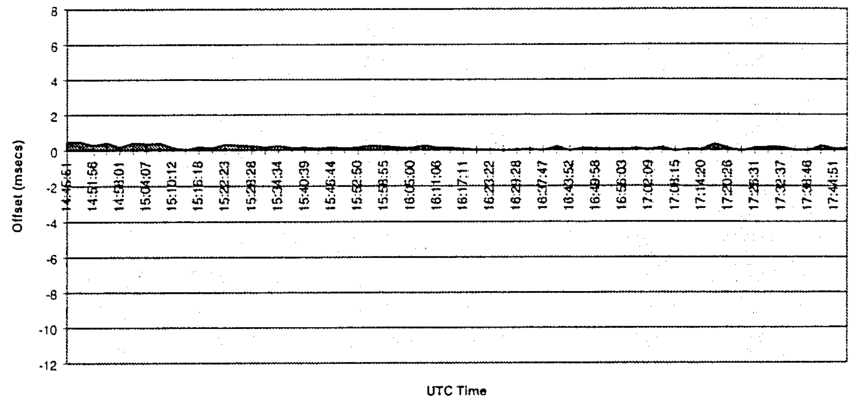

Figure 2: Synchronization of Veda Stratum 2 to Veda Stratum 1

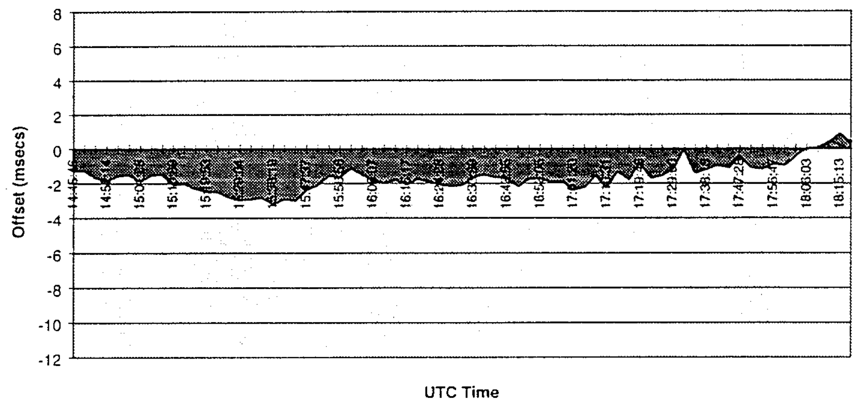

Figure 3: Synchronization of IST Stratum 2 (CGF) to Veda Stratum 1

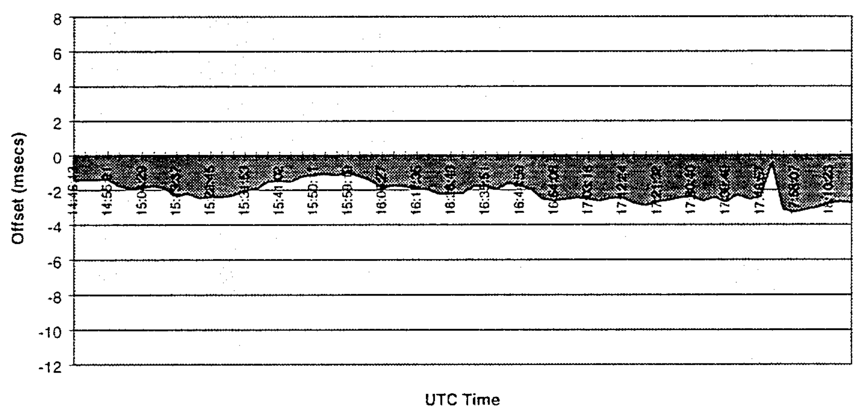

Figure 4: Synchronization of IST Stratum 2 (Logger) to Veda Stratum 1

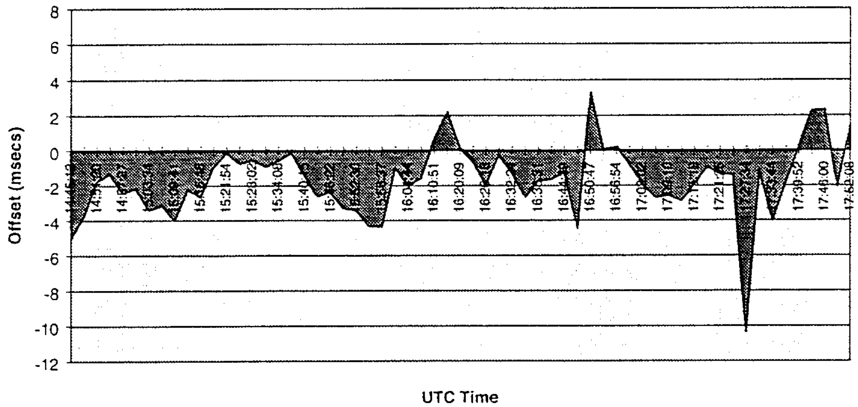

Figure 5: *Synchronization of TNO Stratum 2 (ModSAF) to Veda Stratum 1*

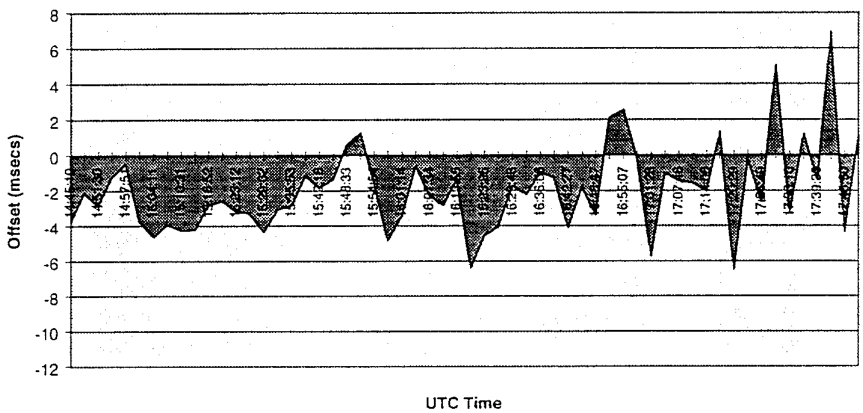

Figure 6: *Synchronization of TNO Stratum 2 (Logger) to Veda Stratum 1*

5. Analysis Methodology

We used the public domain 'tcpdump' utility to log network traffic. Tcpdump prepends a low level time of receipt timestamp to each incoming packet. The simple file and packet headers in the tcpdump format are well suited for subsequent conversion to other standard logger formats, such as the Data Logger Interchange Format (DLIF). Data logged in these experiments were converted to DLIF to allow standard playback and analysis tools to examine the data. Other analysis routines were developed to examine particular aspects of the data. Tcpdump is also capable of filtering based on combinations of UDP port and source addresses. DIS PDU's contain a timestamp in the PDU header as previously discussed. By subtracting PDU timestamp from the time of receipt we obtain the latency. Since the time stamps originate from different machines, there is a prerequisite that all machines are well synchronized. During our experiments IST and TNO generated considerable amounts of Entity State PDU's. Veda generated SIMAN PDU's in smaller numbers. We conducted a series of measurements twice, once during light network load and once during heavy network load, where 'heavy' refers to the capacity of the ISDN routers (64 kbps in our configuration) and not the local networks. To generate background traffic we used the IST PC CGF to generate 32 static entities (each having 2 articulated parts), with a default update interval of 1 second (versus the standard 5 seconds), which provided steady base of approximately 50 kbps. This traffic was issued on a different UDP port and exercise ID to lessen any impact on the simulations.

Tables 1 and 2 show measured average latencies during light and heavy network load. Tables 3 and 4 show maximum latencies during light and heavy network load. IST acted as the network hub during these experiments.
We have also constructed histograms reflecting the distribution of the latencies. Figures 7 and 8 reflect the distribution of latencies to TNO and Veda of DIS PDU's generated by IST during light and heavy network load. Figures 9 and 10 reflect the distribution of the latencies to IST and Veda of DIS PDU's generated by TNO during light and heavy network load. The number of DIS PDU's generated by Veda was too small to produce meaningful histograms, as SIMAN did not constitute a large amount of traffic.
The histograms clearly show the two-legged WAN configuration with IST acting as hub. Figures 7 and 8 show that the latencies from IST to Veda are much smaller than the latencies from IST to TNO. Figures 9 and 10 show that the latencies from TNO to IST are slightly smaller than the latencies from TNO to Veda. Furthermore, the histograms clearly show wider distributions during heavy network traffic.

Table 1: Average latency (ms), light network load (IST acting as hub)

Receiver / Sender	IST	TNO	Veda
IST	1 ms	135 ms	43 ms
TNO	124 ms	2 ms	147 ms
Veda	72 ms	180 ms	46 ms

Table 2: Average latencies during heavy network load

Receiver / Sender	IST	TNO	Veda
IST	2 ms	147 ms	54 ms
TNO	133 ms	1 ms	158 ms
Veda	73 ms	180 ms	45 ms

Table 3: Maximum latencies during light network load

Receiver / Sender	IST	TNO	Veda
IST	2 ms	338 ms	255 ms
TNO	321 ms	32 ms	431 ms
Veda	121 ms	240 ms	100 ms

Table 4: Maximum latencies during heavy network load

Receiver / Sender	IST	TNO	Veda
IST	3 ms	865 ms	787 ms
TNO	539 ms	38 ms	795 ms
Veda	123 ms	229 ms	100 ms

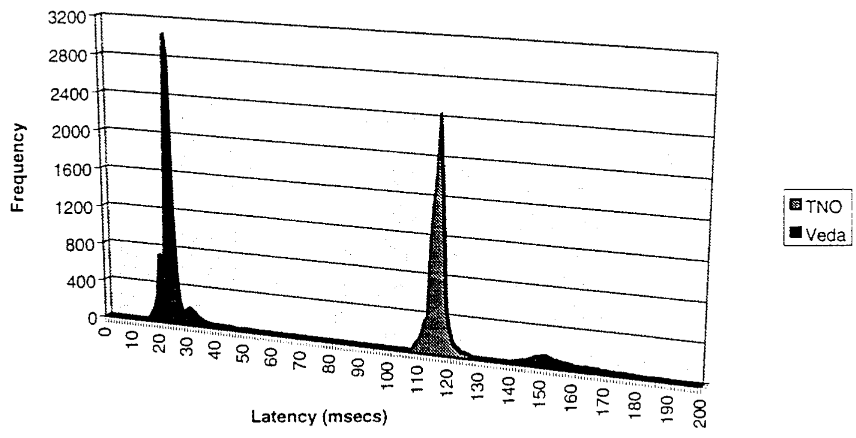

Figure 7: *Latency distribution of IST PDU's to TNO and Veda during light network load*

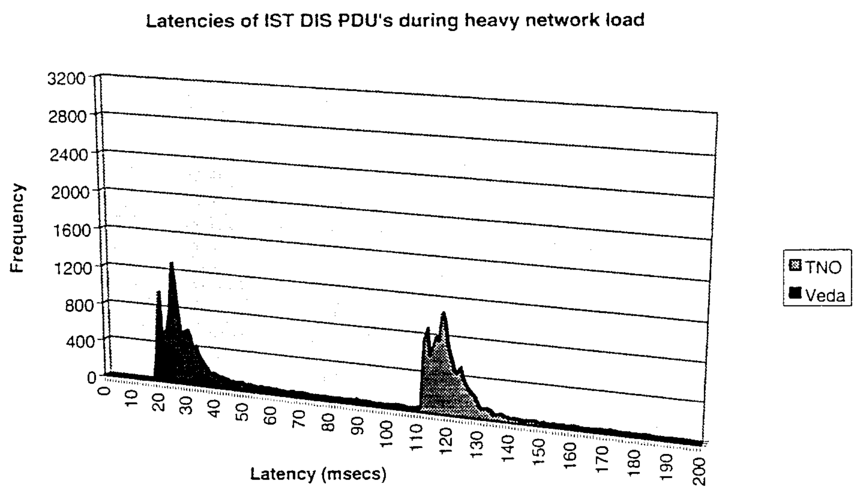

Figure 8: *Latency distribution of IST PDU's to TNO and Veda during heavy network load*

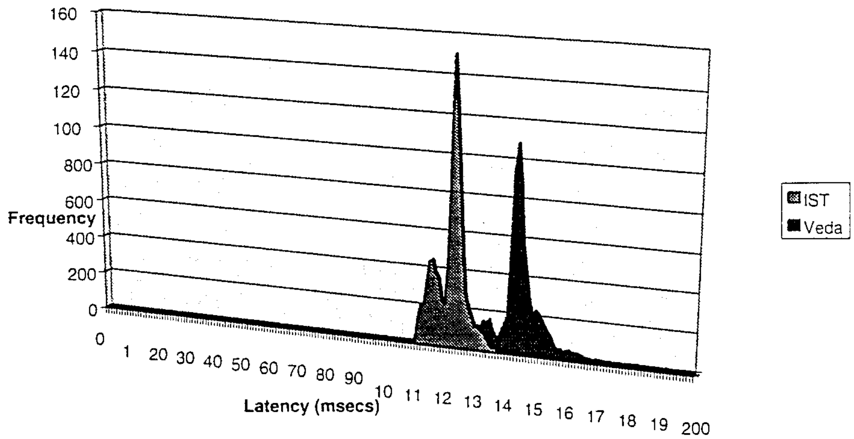

Figure 9: *Latency distribution of TNO PDU's to IST and Veda during light network load*

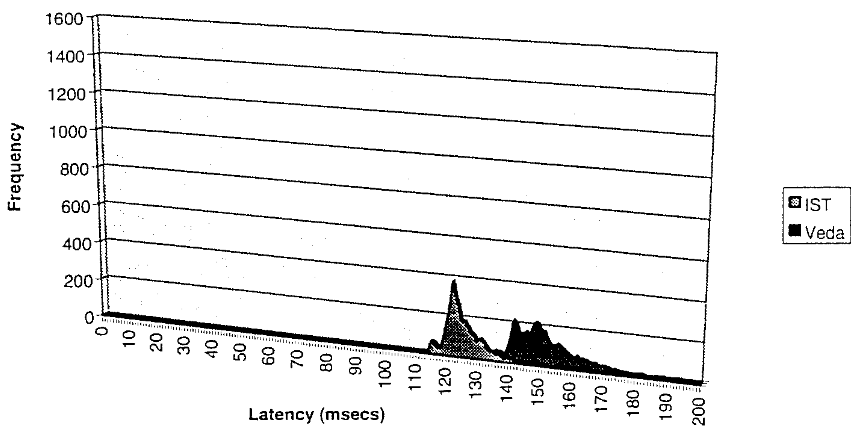

Figure 10: *Latency distribution of TNO PDU's to IST and Veda during heavy network load*

5.1 Latency Observations

During several experiment sessions, scripts were run at the sites that measured the network round-trip delays and clock offsets between the key systems every three minutes. For these measurements we executed both the application program ntpdate and the xntpdc query/control program for the NTP daemon. As previously stated, ntpdate is used to set a host clock but may also be used in debug mode to report statistics without actually setting the clock. It measures round-trip delays and clock offsets as seen at the application layer. Ntpdate transmits eight successive NTP messages to the remote system to compute latency and offset from the NTP message header timestamps. Xntpdc queries the NTP daemon, which reports the most recently computed statistics. Note that the data returned by xntpdc may be somewhat dated (stale), whereas ntpdate computes new data with each call.

Analyzing the moment that NTP packets were seen by tcpdump at the SG1, SG2, and DISSN1 systems, we could derive the minimum, median, and average time spent in the systems, router delays, and ISDN latencies, which are shown in Figure 11 and Table 5.

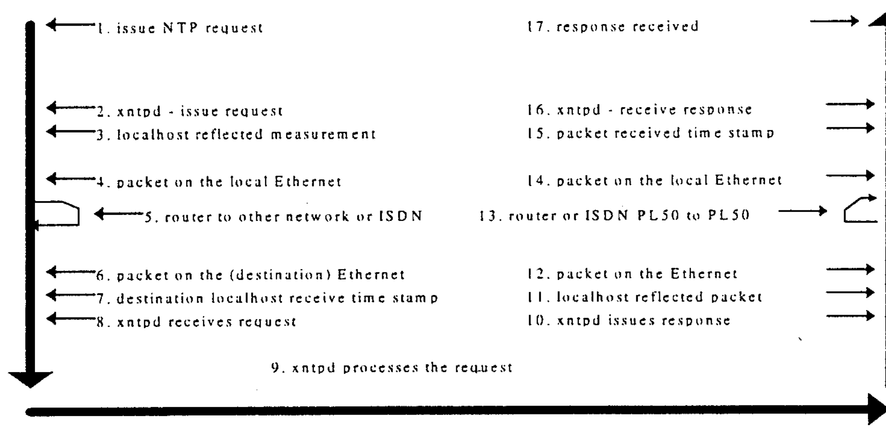

Figure 11: NTP Message Model

Table 5: Timing Observations During DIS Experiments

From-to	Timing	Minimum (ms)	Median (ms)
1 - 3	ntpdate stack time	12.03	12.16
1 - 3, 15 - 17	ntpdate request/receive stacks	24.07	24.32
2 - 16	xntpdc on local Ethernet	1.07	1.37
3 - 4; 14 - 15	to/from Ethernet time	0.11	0.15
4 - 11	ntp-packet processing on SG2 coming from SG2	0.65	0.69
5 or 13	Single direction router delay	0.20	0.30
6 - 12	ntp processing on SG2	0.65	0.69
8 - 10	ntp deamon processing time on dissn1 (Sparcstation 10)	0.05	0.05
5 or 13	VEDA-TNO or IST-TNO and vice-versa ISDN-latency; US site calls TNO	81.0	81.0
5 or 13	TNO-VEDA or TNO-IST and vice-versa ISDN-latency; TNO calls the US	105.6	105.6
5 or 13	IST-VEDA using TNO-FEL as hub (two long legs); one call intiated in the US, one call initiated in The Netherlands	186.0	186.0
5 or 13	IST-VEDA direct ISDN path	13.4	n/m
2-4	DIS packet to LAN	0.3	0.9

One of the observations is the minimum time required to send an ntpdate application packet over the local Ethernet and receive it on another system is 25 ms. As DIS packets are handled at the UDP-interface, the time to the LAN is on the order of one millisecond.

Other observations are ISDN related. As ISDN is a switched telephone-like service, one could expect different routes each time one calls. To our surprise we didn't notice differences in the ISDN path latencies between calls.
However, we noticed a difference in latency of 30% between ISDN connections that were originally dialed from the USA versus those dialed from The Netherlands. Obviously, the Dutch telecom operator KPN Telecom (formerly PTT) uses lines that take a less straight-forward route, presumably at lower cost.

A simple calculation using the positions reported by our GPS-clocks (TNO: 52N 6' 25.2", 4E 19' 36.5" and Veda 28N 35' 58.5", 81W 13' 20.3") shows a distance of 7243 km or 4500.5 miles between The Hague, The Netherlands and Orlando. Given the speed of light, electron speed delay through copper wires and an estimated diversion of 15% of the optimum great circle route, the fastest signal requires 40 ms. This means that the PNO-amplifiers, TDM-systems, switches and so on add up to a 41 ms delay when using ISDN.

Using the same ratio, the ISDN-route when calling from The Netherlands takes a possibly 'scenic' detour of 2500 km. KPN Telecom answered "it is up to us to take the least expensive route. However, the latency difference is so large that we will investigate".

Comparing the direct ISDN routes with measurements made last year using a DSI connection between Orlando and Ramstein, Germany [7] we noticed a DSI-*varying* network latency of 98 ms ±13 ms between Orlando and Ramstein. The fastest *direct* ISDN-link is 20% faster and doesn't show latency variance due to other (not owned) traffic.

Out of curiosity, we did a measurement (using ping) across the Internet between Veda and TNO-FEL which showed an average 480 ms round-trip latency, with individual packet times ranging from 240 ms to over 2000 ms.

5.2 Tools for Collaborative Work

Given the large distance between Florida and The Netherlands as well as the difference in time zones, E-mail and FTP were the obvious vehicles for exchanging analysis, after action reviews, questions and discussions, respectively measured data and log files.

Rob Ripley of Veda suggested the use of Internet Relay Chat (IRC) to communicate between the sites during the conduct of the experiments. This public domain tool turned out be invaluable. With IRC, messages typed in on a IRC client are distributed (relayed) to all 'subscribers' of the same IRC channel. To use this medium, one has to install an IRC client (public domain versions are widely available) on a system connected to the Internet, select a nickname, start IRC and connect to an IRC server on a specific IRC channel. In our case we established and used the #timesync-channel.

As IRC is an open discussion forum, the conversation between the sites can be joined (seen on the screen) in a public forum. Thus it isn't the place to exchange passwords or other sensitive information. IRC proved to be a very efficient (and cheap) communication tool between multiple sites. It allowed for communication among the sites without the expense of voice telephone, and was invaluable in support of the various router and software configurations that took place during setup phases. E-mail messages sometimes require hours to go from sender to receiver, making it a poor mechanism for "real-time" communication. IRC does not suffer from this limitation, as messages are relayed with minimal delay, usually less than a few seconds. During the experiments, multiple independent discussions flashed between the sites on the same chat-channel: prepare, ready, start and stop DIS-exercise messages; discussion about NTP accuracy, ISDN-(re)configuration discussion and discussions on intermediate results. Another advantage of IRC is the ability to save the chat transcripts as a text file. This can provide a valuable hard copy of the session for later review.

6. SIMAN time synchronization

The SIMAN guidance document [12] provides a method for synchronizing an interface unit or an application's "wall clock reference time", but not a host hardware or operating system clock. This method uses the Start/Resume PDU to provide synchronization, using the two time fields within the PDU. A different approach has been utilized in the I/ITSEC DIS demonstrations, in which the Action Request PDU or Set Data PDU was used with a datum ID defined for a "set clock" request. In this method, the time is sent in a Clock Time Record, which provides hours and time past the hour (the latter in the form of a DIS timestamp). Although several Simulation Managers (SM) at I/ITSEC 95 were able to provide this functionality, none of them were expressly synchronized to UTC, making the accuracy of the time questionable. Also, the SIMAN method used for I/ITSEC did not make any allowances for network delay, although the method prescribed in [12] includes a provision for the SM to estimate the latency and convey both time and latency to the destination application. An even greater issue is the accuracy of the clock being synchronized. As experienced in our research, uncorrected clocks can drift by several seconds or more per hour, quickly rendering attempts at synchronization ineffective. What is required for DIS is a system that both synchronizes and stabilizes a system's reference clock, whether in hardware or software. Our experiences with NTP show it to be a more effective and appropriate choice for synchronization than a SIMAN approach. However, the ability to manage time via SIMAN may still be useful in circumstances where NTP is not available for a particular application or site, with the accompanying loss of precision. In the event that a SIMAN method is used, the Simulation Manager must be synchronized to UTC through either a GPS receiver, as used in our experiments, through the use of NTP to a remote time server via the network, or by some other method. However, it cannot be overstated that any scheme of distributing time that does not also address the issue of drift will not provide sufficient accuracy for the use of absolute or relative timestamps.

7. Conclusions

In this report we attempted to show the results of our experiments involving ISDN as a WAN link for DIS, NTP time synchronization, SIMAN time synchronization, absolute versus relative timestamping, and long distance exercise coordination. Some of our experiments answered questions and some brought up more questions. Here is a recap of what we did and what we learned in the process:

- **ISDN:** We demonstrated that ISDN can be an effective WAN-link for DIS exercises. ISDN-lines are reliable, relatively inexpensive, have stable latencies, and are becoming more widely available. ISDN minimum transport latencies are dependent upon which carrier is used to setup the connection. The carrier may choose a route that is much greater than the shortest path in order to use lower cost lines, potentially adding tens of milliseconds to the latency. Taking the large differences in international ISDN tariffs, one should take this possible 'delay' factor in mind. Most of the problems we had regarding the WAN-link involved ISDN router setup, not ISDN as a technology. The Ascend PL50s are inexpensive and easy-to-use ISDN routers, but the software bug we found related to bridging broadcast packets must be fixed by the vendor to facilitate their use in DIS.

- **NTP:** We learned how to set up an NTP stratum-1 server with an attached GPS receiver. We synchronized our computer clocks to UTC (to a certain degree) by using these stratum-1 servers. We discovered that NTP can synchronize computer clocks over a long distance WAN link with only one stratum-1 server available. We documented all the steps necessary to use NTP in a DIS exercise.

- **Clock Drift**: NTP was effective in allowing us to measure and compensate for drift on each system clock. We observed one machine at TNO and a similar machine at IST drifted by 2 seconds each 6-7 minutes, i.e. a drift of over 5000 ppm (> 0.5 %). When using relative timestamping, the difference in time stamps is said to be a constant clock offset plus a variable latency. The minimal or average time stamp difference is maintained. When greater time stamp differences occur, it is assumed to be caused by additional latency, which is compensated for in dead reckoning algorithms. Relative timestamping assumes a constant clock offset, or at least a drift that is negligible compared to the latency variance. After 6-7 minutes, relative timestamping would unjustly compensate for 2 seconds! In such cases ignoring time stamps altogether would be better than using relative timestamping. This supports our conclusions that absolute timestamping is the preferred method that should be used where feasible, and that time of receipt may be acceptable when drift is unknown.

- **SIMAN Time Synchronization:** After gaining appreciation for the complexities involved in synchronizing clocks to UTC using NTP, we

concluded that the SIMAN approach to time synchronization is very limited and should only be used if NTP is not available.

- **Absolute versus Relative Timestamping:** We drew the conclusion that for millisecond accuracy over long haul networks, absolute timestamping is a must. Relative timestamping has limited value due to uncontrollable variables such as clock drift, and in many cases ignoring timestamps all together and using time of receipt may be better than using relative timestamping.

- **Exercise Coordination:** As a big cost saver, we proved that inexpensive (or free) tools such as Internet Relay Chat can be used as a long distance communications medium for exercise coordination.

8. Acknowledgements

The authors of this report would like to thank Andy Cox (IST) and Rob Ripley (Veda) for their collaboration in the experiments and for their efforts in writing a joint paper for the 14th DIS workshop.

The authors would like to thank as well for the installation and tuning work done on NTP and GPS by Ben Shen (Veda), Michael Myjak (IST), Jan van Brussel (TNO-FEL) and Ewald Beekman (TNO-FEL). Invaluable to the project was the ISDN router guru Lex Beijk (TNO-FEL).

The support of our sponsors Ir. O. Hoogesteijn (RNLA/DMKL/T&WO) and Mr. Michael P. Garnsey (U.S. Army STRICOM) is gratefully acknowledged.

9. References

[1] *"It's about Time, it's about Space - Time and Space in DIS"*, Saunders, R, 12th DIS Workshop on Standards for the Interoperability of Distributed Simulations; IST-CF-95-01.1 Orlando, Florida March 1995, paper 12-95-015 page 63.

[2] *Network Time Protocol (Version 3) Specification, Implementation and Analysis*, David Mills, DARPA Network Working Group document RFC-1305, March 1992.

[3] *"The Application of Network Time Protocol (NTP) To Implementing DIS Absolute Timestamp"*, Mitchell Golner and Eytan Pollak, 11th DIS Workshop Paper #94-151, September 1994.

[4] *Notes on xntpd Configuration*, David Mills, document distributed with Xntp version 3.4x, January 14, 1993.

[5] *Introduction to Network Time Synchronization with NTP*, Richard E. Schmidt, U.S. Naval Observatory, document found on the Internet.

[6] *E-mail conversations with Steve Clift of the CSIRO Marine Labs*, Hobart, Australia.

[7] *The use of ISDN communication links for Distributed Interactive Simulations*, H.A.M. Luiijf, .13th DIS Workshop on Standards for the Interoperability of Distributed Simulations; IST-CF-95-02 Orlando, Florida September 1995, paper 95-13-32 page 205.

[8] *"DIS at Nine Gs"*, Swaine, S., Marz, T., 13th DIS Workshop on Standards for the Interoperability of Distributed Simulations; IST-CF-95-02 Orlando, Florida September 1995, paper 95-13-042 page 259.

[9] *"The limitations of interactive behavior for valid Distributed Interactive Simulation"*, Foster, L. and Feldmann, P.; IST-CF-95-02 Orlando, Florida September 1995, paper 95-13-027 page 175.

[10] *"DIS Time"*, Myjak, M., Presentation at the 13th DIS Workshop on Standards for the Interoperability of Distributed Simulations; Florida September 1995.

[11] *"Precision under DIS"*, Katz, A., 12th DIS Workshop on Standards for the Interoperability of Distributed Simulations; IST-CF-95-01.1 Orlando, Florida March 1995, paper 12-95-090 page 531.

[12] *Simulation Management Guidance Document*, Rough Draft, Version 1.1, October 12, 1995.

[13] *"Time Synchronization Experiments"*, Cox, A., van Kampen, R. Luiijf, H.A.M., Ripley, R. , 14th DIS Workshop on Standards for the Interoperability of Distributed Simulations; IST-CF-96-03 Orlando, Florida March 1996, paper 96-14-091, pp 599-613.
A Word for Windows 6.0 version of the paper can be retrieved via http://www.tno.nl/insit/fel/refs

[14] *The use of ISDN communication links for Distributed Interactive Simulations*, H.A.M. Luiijf, A.F. Beijk, TNO-report FEL-95-A024, March 1995.

10. Signature

Ir. W.G. de Jong
Project leader

Ir. H.A.M. Luiijf
Author

Appendix A List of abbreviations

B-channel	64 kbps ISDN bearer channel
CGF	Computer Generated Forces
D-channel	16 kbps ISDN signaling channel
DEA	Data Exchange Agreement
DIS	Distributed Interactive Simulation
DMKL	Directie Materieel Koninklijke Landmacht (RNlA)
DSI	Defense Simulation Internet
FEL	TNO Physics and Electronics Laboratory
GPS	Global Positioning System
I/ITSEC	Interservice/Industry Training Systems and Education Conferences
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
IST	Institute for Simulation and Training, University of Central Florida, Orlando, Florida
kbps	kilobits per second
LAN	Local Area Network
NTP	Network Time Protocol (RFC 1305)
PDU	Protocol Data Unit
RFC	Request For Comment
RNlA	Royal Netherlands Army
SGI	Silicon Graphics Inc.
SIMAN	Simulation Manager
STRICOM	Simulation and Training Command (US Army)
TCP	Transmission Control Program (RFC 793, MIL-STD 1778)
TNO	Netherlands Organization for Applied Scientific Research
UDP	User Datagram Protocol (RFC 768)
USNO	U.S. Naval Observatory
UTC	Coordinated Universal Time
VEDA	Veda Inc, Orlando, Florida
WAN	Wide Area Network

REPORT DOCUMENTATION PAGE
(MOD-NL)

1. DEFENCE REPORT NO (MOD-NL)	2. RECIPIENT'S ACCESSION NO	3. PERFORMING ORGANIZATION REPORT NO
TD96-0060		FEL-96-A078
4. PROJECT/TASK/WORK UNIT NO	**5. CONTRACT NO**	**6. REPORT DATE**
25873	A95KL840 DEA Annex A-94-TN-1529	May 1996
7. NUMBER OF PAGES	**8. NUMBER OF REFERENCES**	**9. TYPE OF REPORT AND DATES COVERED**
34 (incl 1 appendix, excl RDP & distribution list)	14	Final

10. TITLE AND SUBTITLE

Time Synchronization in Distributed Simulations

11. AUTHOR(S)

H.A.M. Luiijf, R. van Kampen

12. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

TNO Physics and Electronics Laboratory, PO Box 96864, 2509 JG The Hague, The Netherlands
Oude Waalsdorperweg 63, 2497 AK The Hague, The Netherlands

13. SPONSORING AGENCY NAME(S) AND ADDRESS(ES)

Ministry of Defense (DMKL/B&B/T&WO)
Binckhorstlaan135, 2516 BA The Hague, The Netherlands

14. SUPPLEMENTARY NOTES

The classification designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret.

15. ABSTRACT (MAXIMUM 200 WORDS (1044 BYTE))

This report presents the results of cooperative research conducted between TNO-FEL and the Institute for Simulation and Training (IST), and Veda, Inc., both located in Orlando, FL. The research focuses on two areas: examining the characteristics of Integrated Services Digital Networks (ISDN), and conducting DIS experiments using absolute timestamps across a Wide Area Network (WAN). Each of these areas are relevant to the DIS community. ISDN is potentially a valuable asset as a primary or backup communications link, offering relatively high bandwith, increasing availability and low costs, whereas the use of absolute time is a fundamental yet somewhat unresolved issue in distributed (interactive) simulation. Our experiences in setting up an absolute time source and in configuring the Network Time Protocol (NTP) are well documented in later sections, which we hope will be seen as a valuable reference. Additionally, references are cited which address the key issues of latency, variance, time synchronization, and dead reckoning, and are highly recommended reading to gain a clearer insight into the issues involved in the use of absolute timestamps in distributed simulations.

16. DESCRIPTORS	IDENTIFIERS
Computerized Simulations Communication Networks Distributed Interactive Simulations ISDN	Electronic Battlefield Network Time Protocol (NTP) Time Synchronization

17a.SECURITY CLASSIFICATION (OF REPORT)	17b.SECURITY CLASSIFICATION (OF PAGE)	17c.SECURITY CLASSIFICATION (OF ABSTRACT)
Ongerubriceerd	Ongerubriceerd	Ongerubriceerd

18. DISTRIBUTION AVAILABILITY STATEMENT	17d.SECURITY CLASSIFICATION (OF TITLES)
Unlimited Distribution	Ongerubriceerd

Distributielijst

1. Bureau TNO Defensieonderzoek
2. Directeur Wetenschappelijk Onderzoek en Ontwikkeling*)
3. HWO-KL
4. HWO-KLu*)
5. HWO-KM*)
6. HWO-CO*)

7 t/m 9. KMA, Bibliotheek

10 t/m 15. DMKL/B&B/T&WO, t.a.v. Ir. O. Hoogesteijn
(13 t/m 15. US AMC/STRICOM, attn. AMSTI - Mr. Gene Wiehagen)

16. DMKL/INFO/SIM, t.a.v. Ing. J.J. Smit
17. Bureau TNO Defensieonderzoek, t.a.v. Ir. W.C. Borawitz
18. Directie TNO-FEL, t.a.v. Dr. J.W. Maas
19. Directie TNO-FEL, t.a.v. Ir. J.A. Vogel, daarna reserve
20. Archief TNO-FEL, in bruikleen aan M&P*)
21. Archief TNO-FEL, in bruikleen aan A.F. Beijk
22. Archief TNO-FEL, in bruikleen aan Ir. J. Bruin
23. Archief TNO-FEL, in bruikleen aan Dr. Ir. A.C.M. Dumay
24. Archief TNO-FEL, in bruikleen aan Ir. W.G. de Jong
25. Archief TNO-FEL, in bruikleen aan Ir. R. van Kampen
26. Archief TNO-FEL, in bruikleen aan Ir. H.A.M. Luiijf
27. Archief TNO-FEL, in bruikleen aan Ir. R.C. van Rijnsoever
28. Documentatie TNO-FEL
29. Reserve

TNO-PML, Bibliotheek**)
TNO-TM, Bibliotheek**)
TNO-FEL, Bibliotheek**)

Indien binnen de krijgsmacht extra exemplaren van dit rapport worden gewenst door personen of instanties die niet op de verzendlijst voorkomen, dan dienen deze aangevraagd te worden bij het betreffende Hoofd Wetenschappelijk Onderzoek of, indien het een K-opdracht betreft, bij de Directeur Wetenschappelijk Onderzoek en Ontwikkeling.

*) Beperkt rapport (titelblad, managementuittreksel, RDP en distributielijst).
**) RDP.